The Tsquery Language

Contents

General Structure

The tsquery language is a query language used to retrieve time-series data from the Cloudera Manager time-series data store. A tsquery has the following structure:

```
SELECT [metric expression] WHERE [predicate list]

examples:

(1) select * where roleType=datanode
Meaning: retrieve time-series for all metrics for all DATANODEs.

(2) select total_cpu_user where roleType=DATANODE
Meaning: retrieve the total_cpu_user metric time-series for all DATANODEs.

(3) select jvm_heap_used_mb / 1024, jvm_heap_committed_mb / 1024 where
category=ROLE and hostname="my host"
Meaning: retrieve the jvm_heap_used_mb metric time-series divided by 1024
and the jvm_heap_committed metric time-series divided by 1024 for all roles
running on the host named "my host".

(4) SELECT jvm_total_threads,jvm_blocked_threads
Meaning: retrieve the jvm_total_threads and jvm_blocked_threads metrics
time-series for all entities which have these two metrics.
```

Each tsquery returns one or more time-series. The (2) tsquery example shown above returns one time-series for each DataNode. A time-series is a stream of metric data points for a specific entity. Each metric data point contains a timestamp and the value of that metric at that timestamp. See the section Entites and Predicates below for details on how modeled by Cloudera Manager.

Multiple tsqueries can be concatenated with semi-colons. The (3) example shown above can be written as:

```
select jvm_heap_used_mb / 1024 where category=ROLE and hostname=myhost;
select jvm_heap_commited_mb / 1024 where category=ROLE and hostname=myhost
```

Tsquery tokens are case insensitive: `Select`, `select` or `SeLeCt` are accepted for `SELECT`. This applies for all tsquery tokens. Tsquery attribute names and most attribute values are also case insensitive. The `displayName` and `entityName` attributes are two whose values are case sensitive.

The metric expression can be replaced with a * (asterisk) as shown in the (1) example above. In that case, all metrics that are applicable for selected entities, such as `DATANODEs` in the (1) example, will be returned.

Filter expressions can be omitted as shown in the (4) example above. In that case, time-series for all entities for which the metrics are appropriate will be returned. For this query you would see the jvm_new_threads metric for NameNodes, DataNodes, TaskTrackers, and so on.

The query `select *` is invalid. For any other query, a maximum of 250 time-series will be returned. This value can be configured in the SCM server settings.

Metric Expression

A metric expression is a comma-delimited list of one or more metric expression statements. A metric expression statement is either the name of a metric collected by Cloudera Manager or a scalar value. For example:

```
jvm_heap_used_mb, cpu_user, 5
```

See "How do I discover which metrics are available for which entities?" in the FAQ below to learn how to discover the metrics collected by Cloudera Manager.

Metric Expression Operators

Metrics expressions support the following binary operators:

- + (plus)
- - (minus)
- * (multiplication)
- / (division)

The following are examples of legal metric expressions:

```
total_cpu_user + total_cpu_system
1000 * (jvm_gc_time_ms / jvm_gc_count)
```

Metric Expression Functions

Metrics expressions support the following functions:

- dt: derivative with negative values. The change of the underlying metric expression, per second.
- dt0: derivative where negative values are skipped (useful for dealing with counter resets). The change of the underlying metric expression, per second.

So the following are all legal metric expressions:

```
dt(jvm_gc_count)
dt0(jvm_gc_time_ms) / 10
```

getHostFact

getHostFact(string factName, double defaultValue). Retrieves a fact about a host, for example:

```
select dt(total_cpu_user) / getHostFact(numCores, 2) where category=HOST
```

The example above will divide the results of dt(total_cpu_user) by the current number of cores for each host. If the number of cores cannot be determined the default will be used, '2'.

getHostFact currently supports one fact, 'numCores'.

Predicate List

A predicate is a 'key operator value' pair in which `key` is a time-series attribute, `value` is a possible value, and `operator` can be either '=' or '`like`'. AND and OR are logical operators that can be used to compose a complex predicate. For example:

```
(1) select * where roleType=DATANODE
Meaning: retrieve all time-series for all metrics for DATANODEs.

(2) select * where roleType=DATANODE or roleType=TASKTRACKER
Meaning: retrieve all time-series for all metrics for DATANODEs or
TASKTRACKERs.

(3) select * where (roleType=DATANODE or roleType=TASKTRACKER) and
hostname=myhost
Meaning: retrieve all time-series for all metrics for DATANODEs or
TASKTRACKERs that are running on host named myhost.

(4) select total_cpu_user where category=role and hostname like "host[0-
3]+.*"
Meaning: retrieve the 'total_cpu_metric' for all hosts with names that
match the regular expression "host[0-3]+.*".
```

The 'like' operator accepts only quoted `values`. `value` can be any regular expression as specified in regular expression constructs in the Java Pattern class documentation.

Here are some of the time-series attributes and their possible values.

Time Series Attribute Name	**Possible values**
roleType	NAMENODE, DATANODE, SECONDARYNAMENODE, JOURNALNODE, MASTER, REGIONSERVER, JOBTRACKER, TASKTRACKER, ACTIVITYMONITOR, SERVICEMONITOR, HOSTMONITOR, EVENTSERVER, ALERTPUBLISHER, REPORTSMANAGER, SERVER, AGENT, IMPALAD, STATESTORE
category	ROLE, DIRECTORY, HOST, FILESYSTEM, SERVICE, NETWORK_INTERFACE, DISK, CLUSTER, FLUME_SOURCE, FLUME_CHANNEL, FLUME_SINK
serviceType	HDFS, HBASE, MAPREDUCE, MGMT, ZOOKEEPER, FLUME, IMPALA
displayName	Use quoted strings to specify localized names or names that include spaces.
hostname	host name
hostId	The hostId is ithe canonical identifier for a host in Cloudera Manager. It must be unique and may not change over time. Often the hostname is used as the hostId.
rackId	rack id, e.g., '/default'
clusterId	The cluster id. To specify a cluster by its name, use filter `'where category=CLUSTER and displayName="[the display name]"'`
serviceName	The service id. To specify a service by its name use filter `'category=SERVICE and displayName="[name]"'`
device	Disk device name, e.g., 'sda'
partition	Partition name, e.g., 'sda1'
mountpoint	mount point name, .e.g., '/var', '/mnt/homes'

Time Series Attribute Name	Possible values
iface	network interface name, e.g., 'eth0'.
componentName	flume component name, .e.g., 'channel1', 'sink1'

The category attribute

The category attribute controls the type of the entities returned by the query. Some metrics are collected for more than one type of entities. For example, total_cpu_user is collected for entities of type HOST and for entities of type ROLE. To retrieve the data for all hosts in your deployment use

```
select total_cpu_user where category=HOST
```

The ROLE category applies to all role types (see roleType attribute above). The SERVICE category applies to all service types (see serviceType attribute above). For example, to retrieve the committed heap for all roles on host1 use

```
select jvm_committed_heap_mb where category=ROLE and hostname="host1"
```

FAQ

How do I compare all disk io for all the datanodes that belong to a specific hdfs service?

```
select bytes_read, bytes_written where roleType=datanode and
serviceName=hdfs1
```

Replace 'hdfs1' with the appropriate service name. You can then facet by "Metric" and compare all datanodes byte_reads and byte_writes metric at once. See the Charting Time-Series Data page for more details about faceting.

When would I use a derivative function?

Some metrics represent a counter, e.g., bytes_read. For such metrics it is sometimes useful to see the rate of change instead of the absolute counter value. Use dt or dt0 derivative functions.

When should I use the dt0 function?

Some metrics, like bytes_read represent a counter that always grows. For such metrics a negative rate means that the counter has been reset (e.g., process restarted, host restarted, etc.). Use dt0 for these metrics.

How do I display a threshold on a chart?

Assume that you want to retrieve the latencies for all disks on your hosts, compare them, and show a threshold on the chart to easily detect outliers. Use the following to retrieve the metrics and the threshold:

```
select service_time, await_time, await_read_time, await_write_time, 50
where category=disk
```

You can then facet the results to be all in one chart. The scalar threshold '50' will also be rendered on the chart. See the Charting Time-Series Data page for more details about faceting.

I get "The query hit the maximum results limit" warning. How do I work around the limit?

There is a limit on the number of results that can be returned by a query. When a query results in more time-series streams than the limit a warning for "partial results" is issued. To circumvent the problem try to reduce the number of metrics you are trying to retrieve. You can also use the `like` operator to limit the query to a subset of entities. For example, instead of

```
select service_time, await_time, await_read_time, await_write_time, 50
where category=disk
```

you can use

```
select service_time, await_time, await_read_time, await_write_time, 50
where category=disk and hostname like "host1[0-9]?.cloudera.com"
```

The latter query will retrieve the disks for only ten hosts.

How do I discover which metrics are available for which entities?

One way to discover which metrics are collected by Cloudera Manager is to look at the `List of Metrics` (on the Charts page next to the metric-type-ahead box) or use the metric-type-ahead box. Another way is to retrieve all metrics for the type of entity you are interested in:

```
select * where roleType=datanode and hostname=host1
```